## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (previously presented): A method for decoding a predetermined code word, wherein said code word comprises a number of positions having different values, comprising the steps of:

providing a processor comprising a central processing unit, memory, an input/output interface, and a data bus connecting said central processing unit to said memory and said input/output interface, said processor decoding a predetermined code word;

determining a calculation rule for a soft-output value for each position of said code word, each said position of said code being correlated with said soft-output value, according to the formula

$$L(U_i|y) = \ln \left( \frac{\sum_{c \in \Gamma^i(+1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)}{\sum_{c \in \Gamma^i(-1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)} \right), \quad \text{for} \quad i = 1, \ldots, K,$$

where

$L(U_i|y)$ is a safety measure (soft output) for the i-th position of the code word to be determined;

$y$ is a demodulation result to be decoded;

$c$ is a code word;

$\Gamma^i(\pm 1)$ are all code words for $u_i = \pm 1$; and

$\sigma^2$ is a variance (channel disturbance);

utilizing a characteristic of a convolutional code, in decoding of said code word, for determining said correlation of said individual positions of said code word from which steps follow of determining states in accordance with a shift register operation, and obtaining a trellis representation from these states;

calculating weights $\mu_q(s)$, for an arbitrary choice of $y \in \mathbb{R}^N$, for nodes $(s, q)$ of said trellis representation by evaluating

$$\mu_q : S \to \mathbb{R},$$

$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right)$$

for $q \in \{1,\ldots,Q\}$.

determining mappings $A_m$ by way of said trellis representation, running through said trellis representation in the natural direction, and calculating the term $A_m$ by

$$A_m(s) = \mu_m(s) \sum_{t \in W(s,V_m)} A_{m-1}(t), \quad \text{for } m \in \mathbb{N}$$

and a starting value

$$A_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases}$$

determining mappings $B_m$ by way of said trellis representation, said trellis representation being run through in opposition to a predetermined direction, and calculating the term $B_m$ by

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s, V_{Q-m+2})} B_{m-1}(t), \quad \text{for} \quad 1 \leq m \leq Q,$$

where

$$B_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases}$$

is determined for terminating the recursion; and

determining terms $A_\alpha^i$ by again running through said trellis representation taking into consideration said terms $A_m$ and $B_m$ already determined, according to a relation

$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t),$$

where $j = \left\lfloor \frac{i-1}{b} \right\rfloor + 1$;

determining $K$ positions of said code word according to

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K.$$

2. (previously presented): The method as claimed in claim 1, wherein said convolutional code has binary state transitions, said method further comprising the steps of:

determining mappings $A_m$ recursively by the equation

$$A_m(s) = \mu_m(s)\left(A_{m-1}\left(\hat{T}(+1, s)\right) + A_{m-1}\left(\hat{T}(-1, s)\right)\right), \quad \text{for} \quad m \in \mathbb{N};$$

determining mappings $B_m$ recursively by the equation

$$B_m(s) = \mu_{Q-m+1}(s)\left(B_{m-1}\left(T(s, +1)\right) + B_{m-1}\left(T(s, -1)\right)\right),$$

for $1 \leq m \leq Q$; and

determining terms $A_\alpha^i$, $i \in \{1,...,K\}$, $\alpha \in \{\pm 1\}$ according to the equation

$$A_\alpha^i(y) = \sum_{s \in S} A_{i-1}(s) B_{Q-i+1}(T(s, \alpha)).$$

3. (previously presented): The method as claimed in claim 1, further comprising the step of:

providing a mobile radio network in which said decoding of a predetermined code word operates.

4. (previously presented): The method as claimed in claim 3, wherein said mobile radio network is a GSM network.

5. (previously presented): The method as claimed in claim 1, wherein said predetermined code word is a concatenated code word, said method further comprising the steps of:

providing said calculated soft-output values as input data of another decoder.

6. (currently amended): An arrangement for decoding a predetermined code word, comprising:

a processor having a central processing unit, memory, an input/output interface, and a data bus connecting said central processing unit to said memory and said input/output interface, said processor ~~being configured for carrying out the method according to claim 1.~~ performing the steps of:

determining a calculation rule for a soft-output value for each position of said code word being correlated with said soft-output value, according to the formula

$$L(U_i|y) = \ln \left( \frac{\sum\limits_{c \in \Gamma^i(+1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)}{\sum\limits_{c \in \Gamma^i(-1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)} \right), \quad \text{for} \quad i = 1, \ldots, K,$$

where

$L(U_i|y)$ is a safety measure (soft output) for the i-th position of the code word to be determined;

$y$ is a demodulation result to be decoded;

$c$ is a code word;

$\Gamma^i(\pm 1)$ are all code words for $u_i = \pm 1$; and

$\sigma^2$ is a variance (channel disturbance);

utilizing a characteristic of a convolutional code, in decoding of said code word, for determining said correlation of said individual positions of said code word from which steps follow of determining states in accordance with a shift register operation, and obtaining a trellis representation from these states;

calculating weights $\mu_q(s)$, for an arbitrary choice of $y \in \mathrm{R}^N$, for nodes $(s, q)$ of said trellis representation by evaluating

$$\mu_q : S \to \mathbb{R},$$

$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right)$$

for $q \in \{1,\ldots,Q\}$.

determining mappings $A_m$ by way of said trellis representation, running through said trellis representation in the natural direction, and calculating the term $A_m$ by

$$A_m(s) = \mu_m(s) \sum_{t \in W(s,V_m)} A_{m-1}(t), \quad \text{for} \quad m \in \mathbb{N}$$

and a starting value

$$A_0(s) = \begin{cases} 1 & : \text{ for } s = s_0, \\ 0 & : \text{ else} \end{cases}$$

determining mappings $B_m$ by way of said trellis representation, said trellis representation being run through in opposition to a predetermined direction, and calculating the term $B_m$ by

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s,V_{Q-m+2})} B_{m-1}(t), \quad \text{for} \quad 1 \leq m \leq Q,$$

where

$$B_0(s) = \begin{cases} 1 & : \text{for } s = s_0, \\ 0 & : \text{else} \end{cases}$$

is determined for terminating the recursion; and

determining terms $A_\alpha^i$ by again running through said trellis representation taking into consideration said terms $A_m$ and $B_m$ already determined, according to a relation

$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s,V_j^i(\alpha))} B_{Q-j+1}(t),$$

where $j = \left\lfloor \frac{i-1}{b} \right\rfloor + 1$;

determining $K$ positions of said code word according to

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K.$$